

Продолжение. Начало в № 5 `2007

Валерий ЗОТОВ
walerry@km.ru

Проектирование блоков синхронизации цифровых устройств, реализуемых на базе модулей DCM в ПЛИС FPGA серии Spartan™-3, с помощью «мастера» Architecture Wizard САПР серии Xilinx ISE

Создание блоков синхронизации цифровых устройств, реализуемых на базе модулей DCM в ПЛИС семейств Spartan-3, Spartan-3L, Spartan-3E и Spartan-3A, с помощью «мастера» Architecture Wizard в соответствии с конфигурацией Cascading in Series with Two DCM_SPs

Конфигурация *Cascading in Series with Two DCM_SPs* применяется, прежде всего, при создании блоков синхронизации, которые должны обеспечивать формирование такого количества выходных сигналов с различными значениями частоты, которое выходит за рамки возможностей модулей синхронизации с конфигурацией *Single DCM_SP*. Использование конфигурации *Cascading in Series with Two DCM_SPs* также эффективно для достижения требуемых значений фазового сдвига сигналов, формируемых делителем и умножителем частоты, ко-

торые не могут быть реализованы в блоках синхронизации с применением одного модуля DCM. Обобщенная структура блоков синхронизации, выполняемых в соответствии с конфигурацией *Cascading in Series with Two DCM_SPs*, показана на рис. 21. Ее основу образуют два модуля DCM, включенных последовательно (в форме каскадного соединения).

В качестве исходного тактового сигнала, который подается на соответствующий вход первого модуля DCM, используется внешний сигнал синхронизации. На вход обратной связи этого модуля DCM в большинстве случаев подается сигнал с частотой исходного тактового сигнала, который формируется на выходе CLK0. Исключением является вариант структуры блока синхронизации, в котором для каскадного соединения используется сигнал, формируемый на одном из выходов CLK2X или CLK2X180. В этом случае на вход обратной связи должен подаваться сигнал с частотой, равной удвоенному значению частоты внешнего сигнала синхронизации.

Исходный тактовый сигнал для второго модуля DCM в блоках синхронизации с конфигурацией *Cascading in Series with Two DCM_SPs* снимается с одного из следующих выходов первого модуля DCM: CLK0, CLK90, CLK180, CLK270, CLK2X, CLK2X180 и CLKDV. Выходы сигналов, формируемые цифровым синтезатором частот DFS, не рекомендуется использовать для каскадного соединения модулей DCM. В качестве сигнала сброса для второго модуля DCM применяется инвертированный сигнал, вырабатываемый на выходе LOCKED первого модуля DCM. Таким образом, сигналы синхронизации на выходах второго модуля DCM будут присутствовать только при совпадении фаз входного и выходного сигналов схемы автоподстройки задержек DLL первого модуля DCM.

Для создания с помощью «мастера» Architecture Wizard блока синхронизации, основанного на каскадном соединении двух модулей DCM, следует после определения названия создаваемых файлов, языка описания HDL, средств синтеза, семейства и типа ПЛИС в диалоговой панели выбора конфигурации *Selection*, показанной на рис. 3 (см. Кит № 5 `2007, стр. 91), выделить строку *Cascading in Series with Two DCM_SPs* и подтвердить затем выбранный вариант нажатием клавиши OK. В результате выполненных действий на экране появится диалоговая панель, которая представлена на рис. 22.

Данная диалоговая панель позволяет выбрать один из доступных выходных сигналов синхронизации первого модуля DCM, который будет использоваться в качестве входного тактового сигнала для второго модуля DCM. Чтобы указать требуемый сигнал, формируемый первым модулем DCM, нужно воспользоваться полем выбора *Select clock to be used for cascading*, которое расположено во встроенной панели *For DCM_SP INST1* (рис. 22). Выпадающий список значений этого поля выбора содержит названия всех выходов мо-

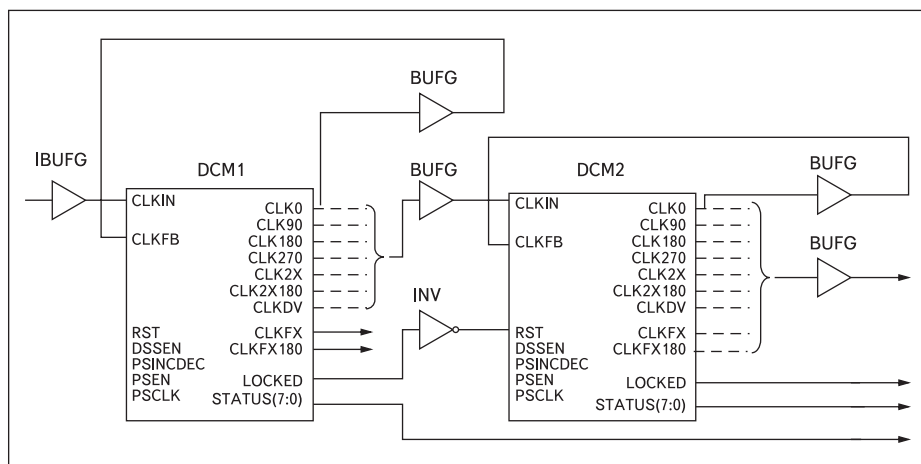


Рис. 21. Обобщенная структура блоков синхронизации, образованная в соответствии с конфигурацией *Cascading in Series with Two DCM_SPs* на базе модулей DCM в ПЛИС семейств Spartan-3, Spartan-3L, Spartan-3E и Spartan-3A

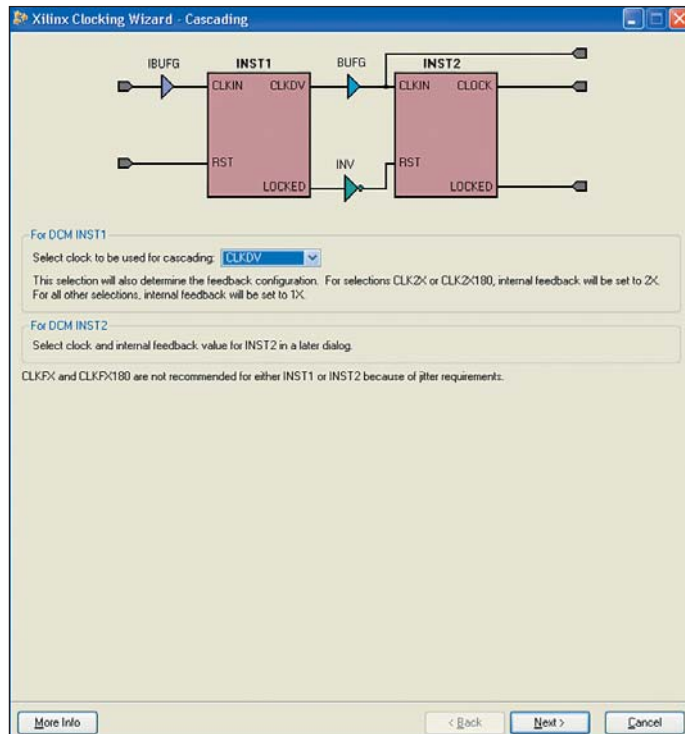


Рис. 22. Вид диалоговой панели выбора сигнала, используемого для каскадного соединения двух модулей DCM в блоках синхронизации с конфигурацией Cascading in Series with Two DCM_SPs

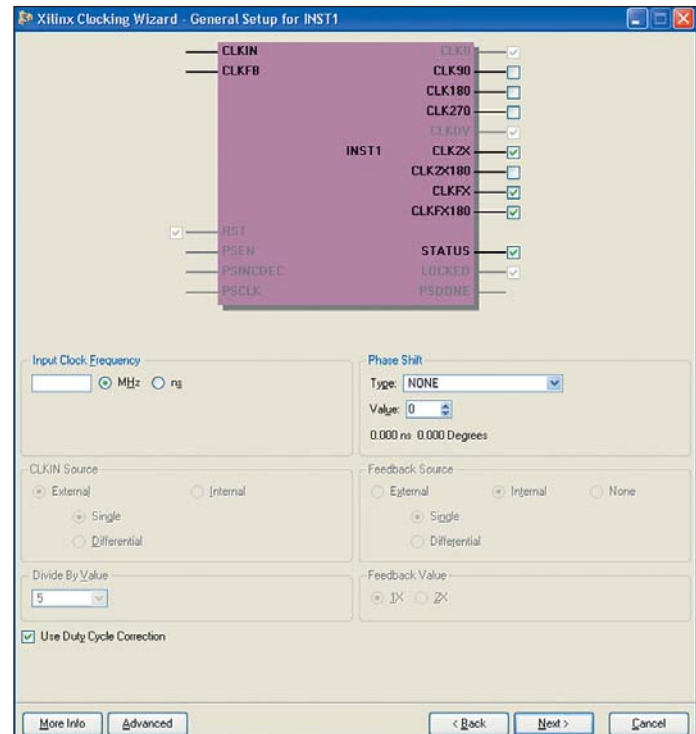


Рис. 23. Диалоговая панель выбора основных параметров первого модуля DCM (INST1) General Setup For INST1 в блоках синхронизации с конфигурацией Cascading in Series with Two DCM_SPs

дуля DCM, которые могут использоваться для каскадного соединения. Дальнейший процесс определения параметров создаваемого блока синхронизации выполняется практически так же, как и при формировании модулей тактирования с конфигурацией *Board Deskew with an Internal Deskew*. Отличия — в диалоговых панелях «мастера» Architecture Wizard с заголовками *General Setup For INST1* и *General Setup For INST2*, которые предназначены для определения главных параметров модулей DCM (INST1 и INST2 соответственно), входящих в состав формируемого блока синхронизации. Вид этих диалоговых панелей при определении параметров блоков синхронизации с конфигурацией *Cascading in Series with Two DCM_SPs* показан на рис. 23, 24.

В диалоговой панели *General Setup For INST1* пользователь должен указать только значение частоты или периода исходного тактового сигнала в поле редактирования, расположенном во встроенной панели *Input Clock Frequency*, а также требуемые дополнительные выходные сигналы первого модуля DCM (INST1) и значение коэффициента деления частоты входного сигнала. В этой же панели, при необходимости, может быть задано значение фазового сдвига выходных сигналов синхронизации, вырабатываемых этим модулем DCM (с помощью поля выбора *Type*, которое находится во встроенной панели *Phase Shift*), а также использование функции коррекции скважности соответствующих тактовых сигналов (с помощью индикатора состояния *Use Duty Cycle Correction*). Все остальные па-

раметры в этой панели устанавливаются автоматически.

Диалоговая панель *General Setup For INST2* позволяет разработчику определить значения

тех же параметров для второго модуля DCM (INST2), за исключением частоты или периода входного сигнала. Значение этого параметра устанавливается автоматически в соответ-

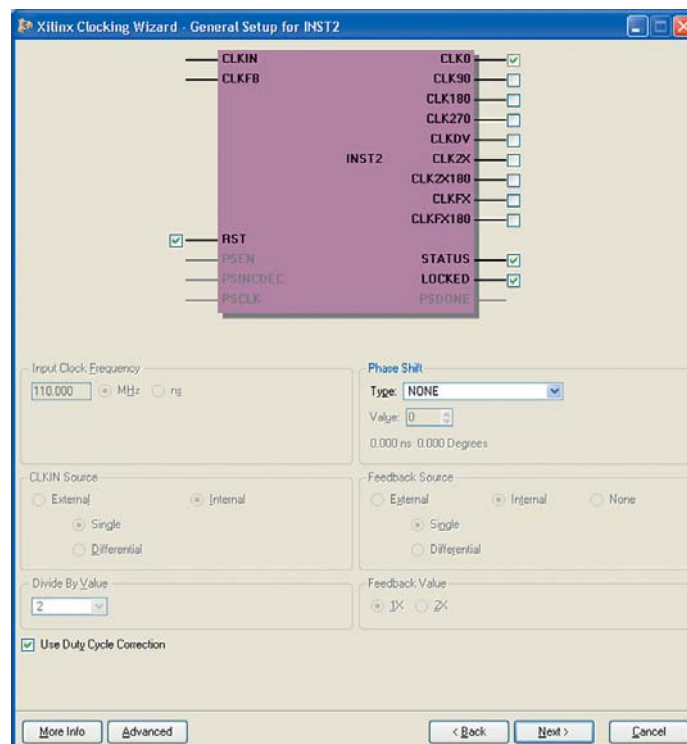


Рис. 24. Диалоговая панель выбора основных параметров второго модуля DCM (INST2) General Setup For INST2 в блоках синхронизации с конфигурацией Cascading in Series with Two DCM_SPs

ствии с выбранным сигналом для каскадного соединения двух модулей DCM, который указан в диалоговой панели «мастера» Architecture Wizard, представленной на рис. 22.

Значения всех заданных параметров модулей DCM отображаются в информационной панели Summary, вид которой приведен на рис. 13 (см. КиТ № 5 `2007, стр. 94). Процесс автоматической генерации описания блока синхронизации с выбранными параметрами запускается нажатием кнопки «Готово» (Finish), которая расположена в нижней части этой информационной панели «мастера».

Пример описания блока синхронизации цифрового устройства с конфигурацией Cascading in Series with Two DCM_SPs, сформированного «мастером» Architecture Wizard

В качестве примера блока синхронизации цифрового устройства, сгенерированного с помощью «мастера» Architecture Wizard в соответствии с конфигурацией *Cascading in Series with Two DCM_SPs*, в данном разделе приводится VHDL-описание модуля `sys_clk_cascading`. Указанный модуль разработан с учетом реализации на основе ПЛИС семейства Spartan-3. Блок синхронизации `sys_clk_cascading` осуществляет формирование совокупности выходных тактовых сигналов, в которую входят:

- четыре сигнала с частотой 70 МГц, различающиеся фазовым сдвигом друг относительно друга, равным четверти, половине и трем четвертям периода этих сигналов (формируются первым модулем DCM);
- сигнал с частотой 35 МГц, получаемый методом двукратного деления частоты исходного тактового сигнала в первом модуле DCM;
- два сигнала с частотой 85 МГц с фазовым сдвигом друг относительно друга на половину периода (с нулевым фазовым сдвигом и фазовым сдвигом, равным половине периода), формируемые цифровым синтезатором частот первого модуля DCM;
- четыре сигнала с частотой 35 МГц, различающиеся фазовым сдвигом друг относительно друга, равным четверти, половине и трем четвертям периода этих сигналов (вырабатываются вторым модулем DCM);
- сигнал с частотой 10 МГц, формируемый делителем частоты входного тактового сигнала с частотой 35 МГц во втором модуле DCM;
- два сигнала с частотой 125 МГц с фазовым сдвигом друг относительно друга на половину периода, вырабатываемые цифровым синтезатором частот второго модуля DCM.

В качестве исходного сигнала синхронизации в блоке `sys_clk_cascading` используется внешний тактовый сигнал с частотой 70 МГц, который подается на соответствующий вход первого модуля DCM. Входным тактовым сигналом для второго модуля DCM является

сигнал, получаемый на выходе делителя частоты первого модуля DCM. Формирование выходных сигналов синхронизации с частотой 70 и 35 МГц выполняется с коррекцией скважности. Выходы этих сигналов подключены к глобальным буферным элементам. Текст сгенерированного VHDL-описания блока синхронизации `sys_clk_cascading` выглядит следующим образом:

```
--Design Name: sys_clk_cascading
--Device: xc3s400-fg320-4
--
-- Module sys_clk_cascading
-- Generated by Xilinx Architecture Wizard
-- Written for synthesis tool: XST
--
library ieee;
use ieee.std_logic_1164.ALL;
use ieee.numeric_std.ALL;
library UNISIM;
use UNISIM.vcomponents.ALL;
--
entity sys_clk_cascading is
    port (
        U1_CLKIN_IN          : in std_logic;
        U1_CLK90_ENABLE_IN  : in std_logic;
        U1_RST_IN           : in std_logic;
        U2_CLK90_ENABLE_IN  : in std_logic;
        U2_CLK180_ENABLE_IN : in std_logic;
        U2_CLK270_ENABLE_IN : in std_logic;
        U1_CLKDV_OUT        : out std_logic;
        U1_CLKFX_OUT        : out std_logic;
        U1_CLKFX180_OUT     : out std_logic;
        U1_CLKIN_IBUFG_OUT  : out std_logic;
        U1_CLK0_OUT         : out std_logic;
        U1_CLK90_OUT        : out std_logic;
        U1_CLK270_OUT       : out std_logic;
        U1_STATUS_OUT       : out std_logic_vector (2 downto 0);
        U2_CLKDV_OUT        : out std_logic;
        U2_CLKFX_OUT        : out std_logic;
        U2_CLKFX180_OUT     : out std_logic;
        U2_CLK0_OUT         : out std_logic;
        U2_CLK2X_OUT        : out std_logic;
        U2_CLK2X180_OUT     : out std_logic;
        U2_CLK90_OUT        : out std_logic;
        U2_CLK180_OUT       : out std_logic;
        U2_CLK270_OUT       : out std_logic;
        U2_LOCKED_OUT       : out std_logic;
        U2_STATUS_OUT       : out std_logic_vector (2 downto 0)
    );
    attribute USELOSKEWLINES : string;
    attribute USELOSKEWLINES of U2_CLKDV_OUT : signal is «»;
    attribute USELOSKEWLINES of U2_CLK2X_OUT : signal is «»;
    attribute USELOSKEWLINES of U2_CLK2X180_OUT : signal is «»;
end sys_clk_cascading;
--
architecture BEHAVIORAL of sys_clk_cascading is
    signal GND1          : std_logic;
    signal STATUS_float  : std_logic_vector (4 downto 0);
    signal STATUS_float1 : std_logic_vector (4 downto 0);
    signal U1_CLKDV_BUF  : std_logic;
    signal U1_CLKFB_IN   : std_logic;
    signal U1_CLKIN_IBUFG : std_logic;
    signal U1_CLK0_BUF   : std_logic;
    signal U1_CLK90_BUF  : std_logic;
    signal U1_CLK180_BUF : std_logic;
    signal U1_LOCKED_INV_IN : std_logic;
    signal U2_CLKFB_IN   : std_logic;
    signal U2_CLKIN_BUF  : std_logic;
    signal U2_CLK0_BUF   : std_logic;
    signal U2_CLK90_BUF  : std_logic;
    signal U2_CLK180_BUF : std_logic;
    signal U2_CLK270_BUF : std_logic;
    signal U2_LOCKED_INV_RST : std_logic;
    -- Period Jitter (unit interval) for block DCM_INST1 = 0.08 UI
    -- Period Jitter (Peak-to-Peak) for block DCM_INST1 = 0.95 ns
    -- Period Jitter (unit interval) for block DCM_INST2 = 0.12 UI
    -- Period Jitter (Peak-to-Peak) for block DCM_INST2 = 0.93 ns
    component DCM
        generic (
            CLK_FEEDBACK : string := «1X»;
            CLKDV_DIVIDE : real := 2.0;
            CLKFX_DIVIDE : integer := 1;
            CLKFX_MULTIPLY : integer := 4;
            CLKIN_DIVIDE_BY_2 : boolean := FALSE;
            CLKIN_PERIOD : real := 10.0;
            CLKOUT_PHASE_SHIFT : string := «NONE»;
            DESKEW_ADJUST : string := «SYSTEM_SYNCHRONOUS»;
            DFS_FREQUENCY_MODE : string := «LOW»;
            DLL_FREQUENCY_MODE : string := «LOW»;
            DUTY_CYCLE_CORRECTION : boolean := TRUE;
        )
    end component DCM
end architecture BEHAVIORAL;
```

```
FACTORY_JF : bit_vector := x«C080»;
PHASE_SHIFT : integer := 0;
STARTUP_WAIT : boolean := FALSE;
DSS_MODE : string := «NONE»
);
port(
    CLKIN          : in std_logic;
    CLKFB          : in std_logic;
    RST            : in std_logic;
    PSEN          : in std_logic;
    PSINCDEC      : in std_logic;
    PSCLK         : in std_logic;
    DSSEN        : in std_logic;
    CLK0           : out std_logic;
    CLK90         : out std_logic;
    CLK180        : out std_logic;
    CLK270        : out std_logic;
    CLKDV         : out std_logic;
    CLK2X         : out std_logic;
    CLK2X180      : out std_logic;
    CLKFX         : out std_logic;
    CLKFX180      : out std_logic;
    STATUS        : out std_logic_vector (7 downto 0);
    LOCKED        : out std_logic;
    PSDONE        : out std_logic
);
end component;
--
component INV
    port (
        I : in std_logic;
        O : out std_logic
    );
end component;
--
component BUFG
    port (
        I : in std_logic;
        O : out std_logic
    );
end component;
--
component IBUFG
    port (
        I : in std_logic;
        O : out std_logic
    );
end component;
--
component BUFGE
    port (
        I : in std_logic;
        CE : in std_logic;
        O : out std_logic
    );
end component;
--
begin
    GND1 <= '0';
    U1_CLKDV_OUT <= U2_CLKIN_BUF;
    U1_CLKIN_IBUFG_OUT <= U1_CLKIN_IBUFG;
    U1_CLK0_OUT <= U1_CLKFB_IN;
    U2_CLK0_OUT <= U2_CLKFB_IN;
    DCM_INST1 : DCM
        generic map(
            CLK_FEEDBACK => «1X»,
            CLKDV_DIVIDE => 2.0,
            CLKFX_DIVIDE => 14,
            CLKFX_MULTIPLY => 17,
            CLKIN_DIVIDE_BY_2 => FALSE,
            CLKIN_PERIOD => 14.2857,
            CLKOUT_PHASE_SHIFT => «NONE»,
            DESKEW_ADJUST => «SYSTEM_SYNCHRONOUS»,
            DFS_FREQUENCY_MODE => «LOW»,
            DLL_FREQUENCY_MODE => «LOW»,
            DUTY_CYCLE_CORRECTION => TRUE,
            FACTORY_JF => x«8080»,
            PHASE_SHIFT => 0,
            STARTUP_WAIT => FALSE
        )
    port map (
        CLKFB=>U1_CLKFB_IN,
        CLKIN=>U1_CLKIN_IBUFG,
        DSSEN=>GND1,
        PSCLK=>GND1,
        PSEN=>GND1,
        PSINCDEC=>GND1,
        RST=>U1_RST_IN,
        CLKDV=>U1_CLKDV_BUF,
        CLKFX=>U1_CLKFX_OUT,
        CLKFX180=>U1_CLKFX180_OUT,
        CLK0=>U1_CLK0_BUF,
        CLK2X=>open,
        CLK2X180=>open,
        CLK90=>U1_CLK90_BUF,
        CLK180=>U1_CLK180_BUF,
```

```

CLK270=>U1_CLK270_OUT,
LOCKED=>U1_LOCKED_INV_IN,
PSDONE=>open,
STATUS(7 downto 3)=>STATUS_float(4 downto 0),
STATUS(2 downto 0)=>U1_STATUS_OUT(2 downto 0)
);
--
DCM_INST2 : DCM
generic map(
  CLK_FEEDBACK => «1X»,
  CLKDV_DIVIDE => 3.5,
  CLKFX_DIVIDE => 7,
  CLKFX_MULTIPLY => 25,
  CLKIN_DIVIDE_BY_2 => FALSE,
  CLKIN_PERIOD => 28.5714,
  CLKOUT_PHASE_SHIFT => «NONE»,
  DESKEW_ADJUST => «SYSTEM_SYNCHRONOUS»,
  DFS_FREQUENCY_MODE => «LOW»,
  DLL_FREQUENCY_MODE => «LOW»,
  DUTY_CYCLE_CORRECTION => TRUE,
  FACTORY_JF => x«8080»,
  PHASE_SHIFT => 0,
  STARTUP_WAIT => FALSE
)
port map (
  CLKFB=>U2_CLKFB_IN,
  CLKIN=>U2_CLKIN_BUF,
  DSSEN=>GND1,
  PCLK=>GND1,
  PSEN=>GND1,
  PSINDEC=>GND1,
  RST=>U2_LOCKED_INV_RST,
  CLKDV=>U2_CLKDV_OUT,
  CLKFX=>U2_CLKFX_OUT,
  CLKFX180=>U2_CLKFX180_OUT,
  CLK0=>U2_CLK0_BUF,
  CLK2X=>U2_CLK2X_OUT,
  CLK2X180=>U2_CLK2X180_OUT,
  CLK90=>U2_CLK90_BUF,
  CLK180=>U2_CLK180_BUF,
  CLK270=>U2_CLK270_BUF,
  LOCKED=>U2_LOCKED_OUT,
  PSDONE=>open,
  STATUS(7 downto 3)=>STATUS_float1(4 downto 0),
  STATUS(2 downto 0)=>U2_STATUS_OUT(2 downto 0)
);
--
INV_INST : INV
port map (
  I=>U1_LOCKED_INV_IN,
  O=>U2_LOCKED_INV_RST
);
--
U1_CLKDV_BUF_INST : BUFV
port map (
  I=>U1_CLKDV_BUF,
  O=>U2_CLKIN_BUF
);
--
U1_CLKIN_IBUF_INST : IBUFV
port map (
  I=>U1_CLKIN_IN,
  O=>U1_CLKIN_IBUFV
);
--
U1_CLK0_BUF_INST : BUFV
port map (
  I=>U1_CLK0_BUF,
  O=>U1_CLKFB_IN
);
--
U1_CLK90_BUF_INST : BUFV
port map (
  CE=>U1_CLK90_ENABLE_IN,
  I=>U1_CLK90_BUF,
  O=>U1_CLK90_OUT
);
--
U2_CLK0_BUF_INST : BUFV
port map (
  I=>U2_CLK0_BUF,
  O=>U2_CLKFB_IN
);
--
U2_CLK90_BUF_INST : BUFV
port map (
  CE=>U2_CLK90_ENABLE_IN,
  I=>U2_CLK90_BUF,
  O=>U2_CLK90_OUT
);
--
U2_CLK180_BUF_INST : BUFV
port map (
  CE=>U2_CLK180_ENABLE_IN,
  I=>U2_CLK180_BUF,
  O=>U2_CLK180_OUT
);

```

```

);
U2_CLK270_BUFV_INST : BUFV
port map (
  CE=>U2_CLK270_ENABLE_IN,
  I=>U2_CLK270_BUF,
  O=>U2_CLK270_OUT
);
--
end BEHAVIORAL;

```

Структура представленного VHDL-описания блока синхронизации `sys_clk_cascading` подобна структуре описания модуля `sys_clk_board_and_int`, приведенного во второй части данной статьи. Основой сформированного описания архитектуры блока синхронизации `sys_clk_cascading` является последовательное (каскадное) соединение двух экземпляров библиотечного компонента DCM, который представляет одноименный модуль кристаллов семейств Spartan-3. Кроме того, в составе сгенерированного структурного описания указанного блока синхронизации применяются и библиотечные компоненты глобальных входных и выходных буферных элементов IBUFV, BUFV, BUFVCE и инвертора INV.

В приведенном описании блока синхронизации `sys_clk_cascading` используется следующая совокупность условных обозначений входных и выходных портов:

- `U1_CLKIN_IN` — вход исходного тактового сигнала;
- `U1_RST_IN` — вход сигнала асинхронного сброса первого модуля DCM;
- `U1_CLK90_ENABLE_IN` — вход разрешения выходного сигнала с частотой 70 МГц со сдвигом фазы на четверть периода, формируемого первым модулем DCM;
- `U2_CLK90_ENABLE_IN` — вход разрешения выходного сигнала с частотой 35 МГц со сдвигом фазы на четверть периода, вырабатываемого вторым модулем DCM;
- `U2_CLK180_ENABLE_IN` — вход разрешения выходного сигнала с частотой 35 МГц со сдвигом фазы на половину периода, формируемого вторым модулем DCM;
- `U2_CLK270_ENABLE_IN` — вход разрешения выходного сигнала с частотой 35 МГц со сдвигом фазы на три четверти периода, вырабатываемого вторым модулем DCM;
- `U1_CLKIN_IBUFV_OUT` — выход исходного тактового сигнала, поступающего со входного буферного элемента;
- `U1_CLK0_OUT` — выход сигнала с частотой 70 МГц с нулевым фазовым сдвигом, вырабатываемый первым модулем DCM;
- `U1_CLK90_OUT` и `U1_CLK270_OUT` — выходы сигналов с частотой 70 МГц с фазовым сдвигом на четверть и три четверти периода соответственно, формируемых первым модулем DCM;
- `U1_CLKFX_OUT` и `U1_CLKFX180_OUT` — выходы сигналов с частотой 85 МГц, отличающихся по фазе на половину периода, которые формируются цифровым синтезатором частот первого модуля DCM;

- `U1_CLKDV_OUT` — выход сигнала с частотой 35 МГц, вырабатываемого делителем частоты первого модуля DCM;
- `U2_CLKDV_OUT` — выход сигнала с частотой 10 МГц, формируемого делителем частоты второго модуля DCM;
- `U2_CLKFX_OUT` и `U2_CLKFX180_OUT` — выходы сигналов с частотой 125 МГц, отличающихся фазовым сдвигом, равным половине периода, которые формируются цифровым синтезатором частот второго модуля DCM;
- `U2_CLK0_OUT` — выход сигнала с частотой 35 МГц с нулевым фазовым сдвигом, который вырабатывается вторым модулем DCM;
- `U2_CLK90_OUT`, `U2_CLK180_OUT` и `U2_CLK270_OUT` — выходы сигналов с частотой 35 МГц с фазовым сдвигом на четверть, половину и три четверти периода соответственно, формируемых вторым модулем DCM;
- `U2_CLK2X_OUT` и `U2_CLK2X180_OUT` — выходы сигналов с частотой 70 МГц, отличающихся фазовым сдвигом, равным половине периода, которые формируются вторым модулем DCM;
- `U2_LOCKED_OUT` — выход сигнала, информирующего о соотношении фаз входного и выходного сигналов схемы автоподстройки задержек DLL второго модуля DCM;
- `U1_STATUS_OUT` и `U2_STATUS_OUT` — выходные шины состояния первого и второго модуля DCM соответственно.

Для определения требуемых значений всех необходимых атрибутов компонентов DCM, применяемых в составе описания блока синхронизации `sys_clk_cascading`, можно добавить в состав файла временных и топологических ограничений проекта разрабатываемого устройства в САПР серии Xilinx ISE следующую совокупность выражений:

```

Net U2_CLKDV_OUT USELOWSKEWLINES;
Net U2_CLK2X_OUT USELOWSKEWLINES;
Net U2_CLK2X180_OUT USELOWSKEWLINES;
INST DCM_INST1 CLK_FEEDBACK = 1X;
INST DCM_INST1 CLKDV_DIVIDE = 2.0;
INST DCM_INST1 CLKFX_DIVIDE = 14;
INST DCM_INST1 CLKFX_MULTIPLY = 17;
INST DCM_INST1 CLKIN_DIVIDE_BY_2 = FALSE;
INST DCM_INST1 CLKIN_PERIOD = 14.2857;
INST DCM_INST1 CLKOUT_PHASE_SHIFT = NONE;
INST DCM_INST1 DESKEW_ADJUST = SYSTEM_SYNCHRONOUS;
INST DCM_INST1 DFS_FREQUENCY_MODE = LOW;
INST DCM_INST1 DLL_FREQUENCY_MODE = LOW;
INST DCM_INST1 DUTY_CYCLE_CORRECTION = TRUE;
INST DCM_INST1 FACTORY_JF = 8080;
INST DCM_INST1 PHASE_SHIFT = 0;
INST DCM_INST1 STARTUP_WAIT = FALSE;
INST DCM_INST2 CLK_FEEDBACK = 1X;
INST DCM_INST2 CLKDV_DIVIDE = 3.5;
INST DCM_INST2 CLKFX_DIVIDE = 7;
INST DCM_INST2 CLKFX_MULTIPLY = 25;
INST DCM_INST2 CLKIN_DIVIDE_BY_2 = FALSE;
INST DCM_INST2 CLKIN_PERIOD = 28.5714;
INST DCM_INST2 CLKOUT_PHASE_SHIFT = NONE;
INST DCM_INST2 DESKEW_ADJUST = SYSTEM_SYNCHRONOUS;
INST DCM_INST2 DFS_FREQUENCY_MODE = LOW;
INST DCM_INST2 DLL_FREQUENCY_MODE = LOW;
INST DCM_INST2 DUTY_CYCLE_CORRECTION = TRUE;
INST DCM_INST2 FACTORY_JF = 8080;
INST DCM_INST2 PHASE_SHIFT = 0;
INST DCM_INST2 STARTUP_WAIT = FALSE;

```

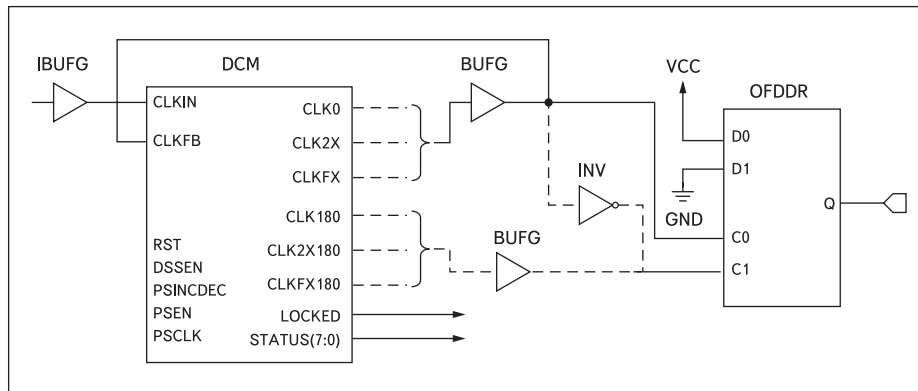


Рис. 25. Обобщенная структура блоков синхронизации, выполняемых в соответствии с конфигурацией Clock Forwarding/Board Deskew на базе модулей DCM в ПЛИС семейств Spartan-3, Spartan-3L, Spartan-3E и Spartan-3A

Формирование блоков синхронизации цифровых устройств, реализуемых на базе модулей DCM в ПЛИС семейств Spartan-3, Spartan-3L, Spartan-3E и Spartan-3A, с помощью «мастера» Architecture Wizard в соответствии с конфигурацией Clock Forwarding/Board Deskew

Блоки синхронизации, создаваемые с помощью «мастера» Architecture Wizard в соответствии с конфигурацией *Clock Forwarding/Board Deskew*, предназначены в первую очередь для формирования совокупности тактовых сигналов, управляющих работой внешних (по отношению к ПЛИС) блоков разрабатываемого устройства. Схема автоподстройки задержки DLL используемого модуля DCM устраняет расхождение фронтов вырабатываемых тактовых сигналов при их распространении по соответствующим проводникам печатной платы проектируемого устройства. Обобщенная структура блоков синхронизации с конфигурацией *Clock Forwarding/Board Deskew* представлена на рис. 25. В ее состав, кроме модуля DCM, входит выходной регистр, выбираемый разработчиком.

В процессе разработки блоков синхронизации с конфигурацией *Clock Forwarding/Board Deskew* необходимо выполнить практически ту же последовательность операций, что и при формировании модулей тактирования с конфигурацией *Single DCM_SP*, которая была подробно изложена в первой части данной статьи. Поэтому далее в настоящем разделе будут рассмотрены только отличия этого процесса.

Для создания блока синхронизации с конфигурацией *Clock Forwarding/Board Deskew* следует в диалоговой панели *Selection*, показанной на рис. 3, выбрать строку, в которой указано название данной конфигурации. Вслед за этим открывается диалоговая панель, которая позволяет выбрать тип выходного регистра, подключаемого к выходу контролируемого сигнала модуля DCM, и сигналы,

которые подаются на его тактовые входы. Вид указанной диалоговой панели определяется применяемым типом выходного регистра и выбранной парой сигналов для его тактирования. На рис. 26 приведено изображение этой диалоговой панели при использовании выходного триггера с асинхронным сбросом, асинхронной установкой и входом разрешения тактового сигнала OFDDRCPЕ, который управляется одним выходным сигналом модуля DCM.

Функцию выходного регистра в формируемом блоке синхронизации выполняют выходные триггеры с удвоенной скоростью записи данных (DDR), которые тактируются парой сигналов синхронизации с фазовым сдвигом друг относительно друга, равным

половине периода. Разработчику предоставляется возможность выбора одного из двух вариантов формирования тактовых сигналов для выходного DDR-триггера. В первом варианте в качестве тактовых сигналов используются два противофазных сигнала, вырабатываемые модулем DCM. Во втором варианте один сигнал синхронизации выходного DDR-триггера поступает с выхода модуля DCM через глобальный буферный элемент, а другой получается в результате инверсии первого сигнала, которая осуществляется с помощью дополнительного инвертора, включаемого в состав создаваемого блока синхронизации.

Тип выходного триггера, подключаемого к соответствующему выходу модуля DCM, указывается в поле выбора *DDR flip-flop selection* (рис. 26). Для включения в состав формируемого блока синхронизации DDR-триггера с асинхронным сбросом, асинхронной установкой, входом разрешения тактового сигнала и тристабильным выходом следует в выпадающем списке этого поля выбрать строку *OFDDRCPЕ*. Вид диалоговой панели при выборе этого варианта выходного DDR-триггера, тактируемого одним выходным сигналом модуля DCM, представлен на рис. 27.

Чтобы установить выходной DDR-триггер с синхронным сбросом, синхронной установкой и входом разрешения тактового сигнала, нужно в поле выбора *DDR flip-flop selection* указать вариант *OFDDRSE*. На рис. 28 представлен вид диалоговой панели при использовании данного типа выходного триггера,

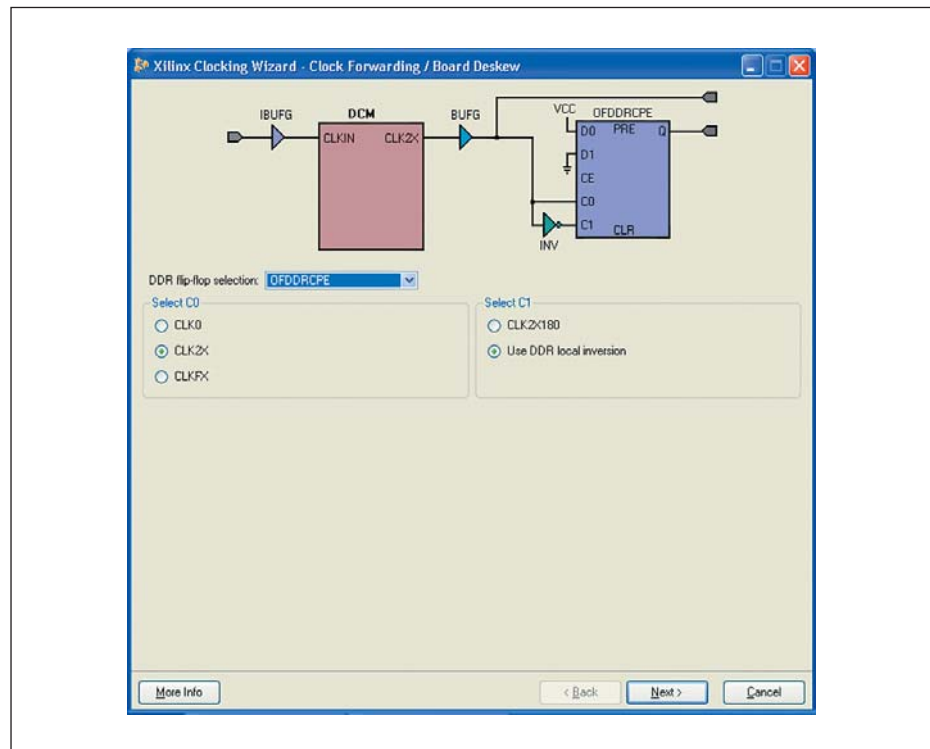


Рис. 26. Вид диалоговой панели, предназначенной для выбора выходного триггера в блоке синхронизации с конфигурацией Clock Forwarding/Board Deskew и способа его управления, при использовании DDR-триггера OFDDRCPЕ, который тактируется одним выходным сигналом модуля DCM

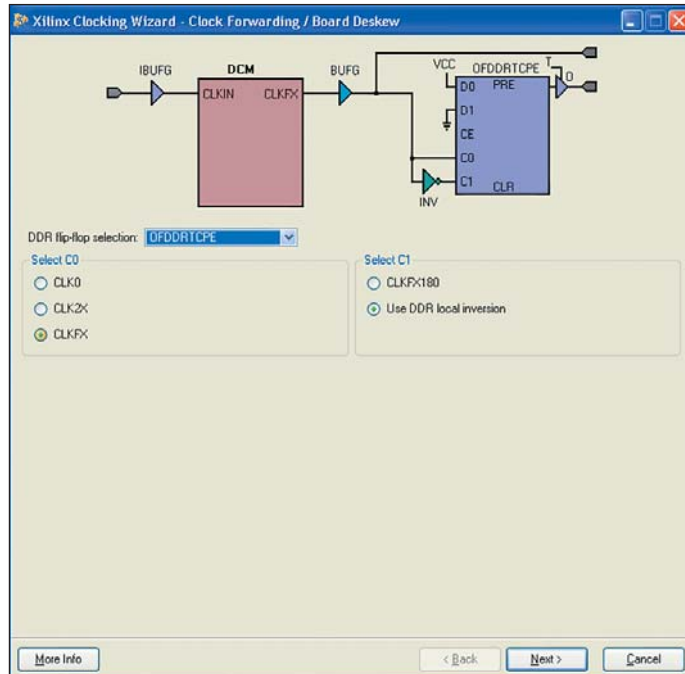


Рис. 27. Вид диалоговой панели, предназначенной для выбора выходного триггера в блоке синхронизации с конфигурацией Clock Forwarding/Board Deskew и способа его управления, при использовании DDR-триггера OFDDRTPCE, который тактируется одним выходным сигналом модуля DCM

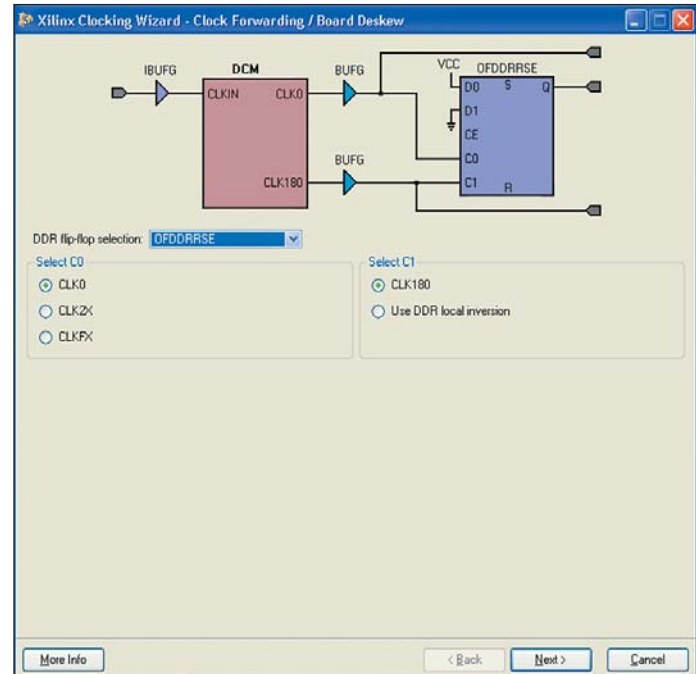


Рис. 28. Вид диалоговой панели, предназначенной для выбора выходного триггера в блоке синхронизации с конфигурацией Clock Forwarding/Board Deskew и способа его управления, при использовании DDR-триггера OFDDRTRSE, который тактируется двумя противофазными выходными сигналами модуля DCM

синхронизируемого двумя противофазными сигналами, которые вырабатываются модулем DCM.

Если на выходе формируемого блока синхронизации необходимо включить DDR-триггер с синхронным сбросом, синхронной установкой, входом разрешения тактового сигнала и тристабильным выходом, то в вы-

падающем списке поля выбора *DDR flip-flop selection* нужно выбрать строку *OFDDRTRSE*. На рис. 29 показан вид диалоговой панели, предназначенной для выбора выходного триггера и способа его управления, в случае использования данного типа триггера, тактируемого двумя противофазными выходными сигналами модуля DCM.

Для генерации блока синхронизации с конфигурацией *Clock Forwarding/Board Deskew* без выходного триггера в поле выбора *DDR flip-flop selection* должно быть указано значение *None*. В этом случае диалоговая панель выбора выходного триггера и способа его управления приобретает вид, показанный на рис. 30.

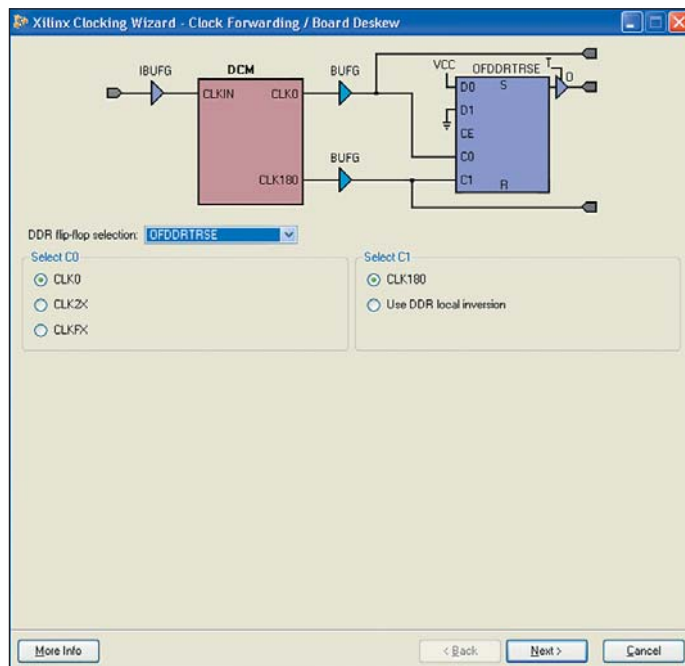


Рис. 29. Вид диалоговой панели, предназначенной для выбора выходного триггера в блоке синхронизации с конфигурацией Clock Forwarding/Board Deskew и способа его управления, при использовании DDR-триггера OFDDRTRSE, который тактируется двумя противофазными выходными сигналами модуля DCM

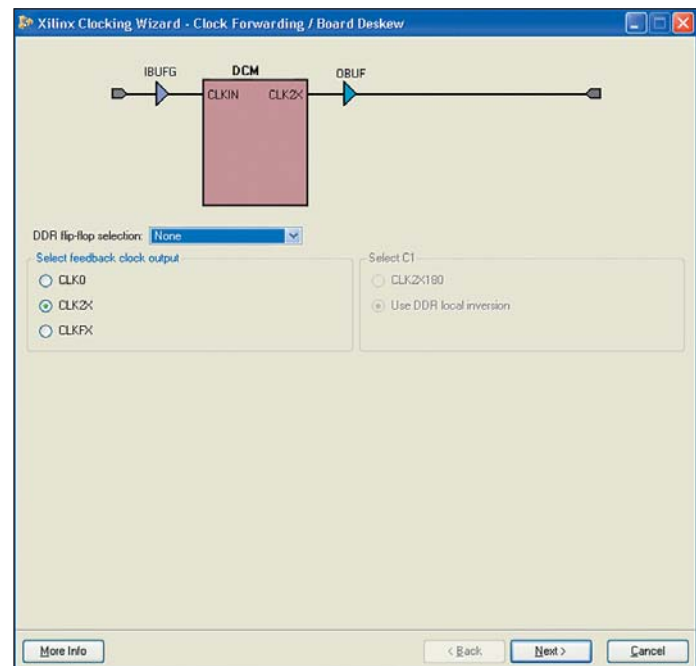


Рис. 30. Вид диалоговой панели, предназначенной для выбора выходного регистра в блоке синхронизации с конфигурацией Clock Forwarding/Board Deskew и способа его управления, при отсутствии выходного триггера

После выбора типа выходного триггера нужно указать сигнал, вырабатываемый модулем DCM, который будет использоваться в качестве сигнала обратной связи для этого модуля. Данный сигнал через глобальный буферный элемент будет подаваться на тактовый вход C0 выходного DDR-триггера. Если выходной триггер отсутствует, то указанный сигнал будет транслироваться через выходной буферный элемент к внешним (по отношению к ПЛИС) компонентам разрабатываемого устройства. Для выбора соответствующего выходного сигнала модуля DCM предназначена группа кнопок с зависимой фиксацией *Select C0* (рис. 26–30). По умолчанию в нажатом состоянии зафиксирована кнопка *CLK0*. При этом на вход C0 выходного триггера или вход выходного буферного элемента будет подаваться сигнал, вырабатываемый на выходе CLK0 модуля DCM, с частотой исходного (внешнего) сигнала синхронизации. Если для этой же цели предполагается использовать сигнал с удвоенным значением частоты исходного сигнала синхронизации, формируемый на выходе CLK2X модуля DCM, то нужно установить в нажатое положение кнопку *CLK2X*. Чтобы выбрать для выполнения той же функции выходной сигнал цифрового синтезатора частот, следует переключить в нажатое состояние кнопку *CLKFX*.

Выбор варианта формирования второго (противофазного) тактового сигнала для выходного DDR-триггера осуществляется с помощью группы кнопок с зависимой фиксацией *Select C1* (рис. 26–29). При нажатой кнопке *CLK180* на вход C1 выходного DDR-триггера будет подаваться сигнал, формируемый на соответствующем выходе модуля DCM с фазовым сдвигом относительно исходного сигнала синхронизации, равным половине периода этого сигнала. Чтобы выбрать вариант формируемого блока синхронизации с использованием выходного DDR-триггера, для тактирования которого применяется один выходной сигнал модуля DCM и дополнительный инвертор, следует зафиксировать в нажатом положении кнопку *Use DDR local inversion*. Если в создаваемом блоке синхронизации выходной триггер отсутствует, то группа кнопок *Select C1* находится в недоступном состоянии.

После завершения процедуры выбора выходного триггера в создаваемом блоке синхронизации с конфигурацией *Clock Forwarding/Board Deskew* и способа формирования его тактовых сигналов последовательно выполняются операции определения значений основных и дополнительных параметров используемого модуля DCM, выбора буферных регистров и указания значений параметров цифрового синтезатора частот. Эти операции осуществляются с помощью соответствующих диалоговых панелей *General Setup*,

Advanced, *Clock Buffers* и *Clock Frequency Synthesizer*, которые были рассмотрены при описании процесса разработки блоков синхронизации с конфигурацией *Single DCM_SP* (рис. 4–6, 12). Активизация процесса автоматической генерации описания создаваемого модуля выполняется так же, как и при создании блоков синхронизации с конфигурацией *Single DCM_SP*.

Пример блока синхронизации цифрового устройства, созданного с помощью «мастера» Architecture Wizard в соответствии с конфигурацией Clock Forwarding/Board Deskew

Примером блока синхронизации с конфигурацией *Clock Forwarding/Board Deskew* является модуль *sys_clk_forwarding_board*, который разработан для реализации на базе ПЛИС семейства Spartan-3E. В качестве исходного тактового сигнала в этом блоке используется внешний сигнал синхронизации с частотой 90 МГц. На выходах данного модуля формируются следующие тактовые сигналы:

- четыре сигнала с частотой 90 МГц, различающиеся фазовым сдвигом друг относительно друга, равным четверти, половине и трем четвертям периода этих сигналов;
- сигнал с частотой 15 МГц, получаемый методом деления частоты исходного тактового сигнала на шесть;
- два сигнала с частотой 180 МГц с фазовым сдвигом друг относительно друга на половину периода (с нулевым фазовым сдвигом и фазовым сдвигом, равным половине периода), формируемые путем двукратного умножения частоты;
- два сигнала с частотой 135 МГц, вырабатываемые цифровым синтезатором частот, с нулевым фазовым сдвигом и фазовым сдвигом, равным половине периода.

В качестве выходного регистра в состав сформированного блока синхронизации включен DDR-триггер с синхронным сбросом, синхронной установкой, входом разрешения тактового сигнала и тристабильным выходом. Выходные сигналы с частотой 180 и 135 МГц с нулевым фазовым сдвигом транслируются через глобальные буферные элементы BUFG. В цепях выходных сигналов с частотой 90 МГц с нулевым фазовым сдвигом и 135 МГц с фазовым сдвигом, равным половине периода, применяются глобальные буферные элементы с входом разрешения синхронизации BUFGCE. В созданном блоке синхронизации используется внешний сигнал обратной связи с частотой исходного тактового сигнала.

Текст VHDL-описания блока синхронизации *sys_clk_forwarding_board*, сгенерированного «мастером» Architecture Wizard, имеет следующий вид:

```
--Design Name: sys_clk_forwarding_board
--Device: xc3s500e-fg320-4
--
-- Module sys_clk_forwarding_board
-- Generated by Xilinx Architecture Wizard
-- Written for synthesis tool: XST
--
library ieee;
use ieee.std_logic_1164.ALL;
use ieee.numeric_std.ALL;
library UNISIM;
use UNISIM.Vcomponents.ALL;
--
entity sys_clk_forwarding_board is
port (
    CE_IN          : in std_logic;
    CLKFB_IN       : in std_logic;
    CLKFX180_ENABLE_IN : in std_logic;
    CLKIN_IN       : in std_logic;
    CLK90_ENABLE_IN : in std_logic;
    RST_IN         : in std_logic;
    R_IN           : in std_logic;
    S_IN           : in std_logic;
    T_IN           : in std_logic;
    CLKDV_OUT      : out std_logic;
    CLKFX_OUT      : out std_logic;
    CLKFX180_OUT   : out std_logic;
    CLKIN_IBUFG_OUT : out std_logic;
    CLK0_OUT       : out std_logic;
    CLK2X_OUT      : out std_logic;
    CLK2X180_OUT   : out std_logic;
    CLK90_OUT      : out std_logic;
    CLK180_OUT     : out std_logic;
    CLK270_OUT     : out std_logic;
    DDR_CLKFX_OUT  : out std_logic;
    LOCKED_OUT     : out std_logic;
    STATUS_OUT     : out std_logic_vector (2 downto 0)
);
end sys_clk_forwarding_board;
--
architecture BEHAVIORAL of sys_clk_forwarding_board is
    signal CLKFB_IBUFG : std_logic;
    signal CLKFX_BUF   : std_logic;
    signal CLKFX_INV_IN : std_logic;
    signal CLKFX180_BUF : std_logic;
    signal CLKIN_IBUFG : std_logic;
    signal CLK0_BUF    : std_logic;
    signal CLK90_BUF   : std_logic;
    signal C0_IN       : std_logic;
    signal GND1        : std_logic;
    signal STATUS_float : std_logic_vector (4 downto 0);
    signal VCC1        : std_logic;
--
    component IBUFG
    port (
        I : in std_logic;
        O : out std_logic
    );
    end component;
--
    component BUFG
    port (
        I : in std_logic;
        O : out std_logic
    );
    end component;
--
    component BUFGCE
    port (
        I : in std_logic;
        CE : in std_logic;
        O : out std_logic
    );
    end component;
--
    -- Period Jitter (unit interval) for block DCM_SP_INST = 0.10 UI
    -- Period Jitter (Peak-to-Peak) for block DCM_SP_INST = 0.71 ns
    component DCM_SP
    generic (
        CLK_FEEDBACK : string := «1X»;
        CLKDV_DIVIDE : real := 2.0;
        CLKFX_DIVIDE : integer := 1;
        CLKFX_MULTIPLY : integer := 4;
        CLKIN_DIVIDE_BY_2 : boolean := FALSE;
        CLKIN_PERIOD : real := 10.0;
        CLKOUT_PHASE_SHIFT : string := «NONE»;
        DESKEW_ADJUST : string := «SYSTEM_SYNCHRONOUS»;
        DFS_FREQUENCY_MODE : string := «LOW»;
        DLL_FREQUENCY_MODE : string := «LOW»;
        DUTY_CYCLE_CORRECTION : boolean := TRUE;
        FACTORY_JF : bit_vector := x«C080»;
    );
end architecture;
```

```

PHASE_SHIFT : integer := 0;
STARTUP_WAIT : boolean := FALSE;
DSS_MODE : string := «NONE»
);
port (
  CLKIN      : in std_logic;
  CLKFB      : in std_logic;
  RST        : in std_logic;
  PSEN       : in std_logic;
  PSINCDEC   : in std_logic;
  PCLK       : in std_logic;
  DSSEN      : in std_logic;
  CLK0       : out std_logic;
  CLK90      : out std_logic;
  CLK180     : out std_logic;
  CLK270     : out std_logic;
  CLKDV      : out std_logic;
  CLK2X      : out std_logic;
  CLK2X180   : out std_logic;
  CLKFX      : out std_logic;
  CLKFX180   : out std_logic;
  STATUS     : out std_logic_vector (7 downto 0);
  LOCKED     : out std_logic;
  PSDONE     : out std_logic
);
end component;
--
component INV
  port (
    I : in std_logic;
    O : out std_logic
  );
end component;
--
component OFDDRTRSE
  port (
    S : in std_logic;
    R : in std_logic;
    T : in std_logic;
    CE : in std_logic;
    D0 : in std_logic;
    D1 : in std_logic;
    C0 : in std_logic;
    C1 : in std_logic;
    O : out std_logic
  );
end component;
--
begin
  GND1 <= '0';
  VCC1 <= '1';
  CLKFX_OUT <= C0_IN;
  CLKIN_IBUFG_OUT <= CLKIN_IBUFG;
  --
  CLKFB_IBUFG_INST : IBUFG
  port map (
    I => CLKFB_IN,
    O => CLKFB_IBUFG
  );
  --
  CLKFX_BUF_INST : BUFG
  port map (
    I => CLKFX_BUF,
    O => C0_IN
  );
  --
  CLKFX180_BUF_INST : BUFGCE
  port map (
    CE => CLKFX180_ENABLE_IN,
    I => CLKFX180_BUF,
    O => CLKFX180_OUT
  );
  --
  CLKIN_IBUFG_INST : IBUFG
  port map (
    I => CLKIN_IN,
    O => CLKIN_IBUFG
  );
  --
  CLK0_BUF_INST : BUFG
  port map (
    I => CLK0_BUF,
    O => CLK0_OUT
  );
  --
  CLK90_BUF_INST : BUFGCE
  port map (
    CE => CLK90_ENABLE_IN,
    I => CLK90_BUF,
    O => CLK90_OUT
  );
  --

```

```

DCM_SP_INST : DCM_SP
generic map(
  CLK_FEEDBACK => «1X»,
  CLKDV_DIVIDE => 6.0,
  CLKFX_DIVIDE => 2,
  CLKFX_MULTIPLY => 3,
  CLKIN_DIVIDE_BY_2 => FALSE,
  CLKIN_PERIOD => 11.1111,
  CLKOUT_PHASE_SHIFT => «NONE»,
  DESKEW_ADJUST => «SYSTEM_SYNCHRONOUS»,
  DFS_FREQUENCY_MODE => «LOW»,
  DLL_FREQUENCY_MODE => «LOW»,
  DUTY_CYCLE_CORRECTION => TRUE,
  FACTORY_JF => «C080»,
  PHASE_SHIFT => 0,
  STARTUP_WAIT => FALSE
)
port map (
  CLKFB => CLKFB_IBUFG,
  CLKIN => CLKIN_IBUFG,
  DSSEN => GND1,
  PCLK => GND1,
  PSEN => GND1,
  PSINCDEC => GND1,
  RST => RST_IN,
  CLKDV => CLKDV_OUT,
  CLKFX => CLKFX_BUF,
  CLKFX180 => CLKFX180_BUF,
  CLK0 => CLK0_BUF,
  CLK2X => CLK2X_OUT,
  CLK2X180 => CLK2X180_OUT,
  CLK90 => CLK90_BUF,
  CLK180 => CLK180_OUT,
  CLK270 => CLK270_OUT,
  LOCKED => LOCKED_OUT,
  PSDONE => open,
  STATUS(7 downto 3) => STATUS_float(4 downto 0),
  STATUS(2 downto 0) => STATUS_OUT(2 downto 0);
  --
  INV_INST : INV
  port map (
    I => C0_IN,
    O => CLKFX_INV_IN
  );
  --
  OFDDRTRSE_INST : OFDDRTRSE
  port map (
    CE => CE_IN,
    C0 => C0_IN,
    C1 => CLKFX_INV_IN,
    D0 => VCC1,
    D1 => GND1,
    R => R_IN,
    S => S_IN,
    T => T_IN,
    O => DDR_CLKFX_OUT
  );
  --
end BEHAVIORAL;

```

Основным компонентом приведенного выше структурного описания архитектуры блока синхронизации `sys_clk_forwarding_board` является библиотечный элемент `DCM_SP`. Кроме того, в составе сформированного описания блока `sys_clk_forwarding_board` используются библиотечные компоненты глобальных входных и выходных буферных элементов `IBUFG`, `BUFG`, `BUFGCE`, выходного `DDR`-триггера `OFDDRTRSE` и инвертора `INV`.

Система условных обозначений интерфейсных портов VHDL-описания блока синхронизации `sys_clk_forwarding_board` выглядит следующим образом:

- `CE_IN` — вход разрешения тактового сигнала, формируемого выходным `DDR`-триггером;
- `CLKFB_IN` — вход сигнала внешней обратной связи модуля `DCM`;
- `CLKFX180_ENABLE_IN` — вход разрешения выходного сигнала с частотой 135 МГц со сдвигом фазы на половину периода;

- `CLKIN_IN` — вход исходного тактового сигнала с частотой 90 МГц;
- `CLK0_ENABLE_IN` — вход разрешения выходного сигнала с частотой 90 МГц с нулевым фазовым сдвигом;
- `RST_IN` — вход сигнала асинхронного сброса модуля `DCM`;
- `R_IN` — вход сигнала синхронного сброса выходного `DDR`-триггера;
- `S_IN` — вход сигнала синхронной установки выходного `DDR`-триггера;
- `T_IN` — вход сигнала управления состоянием (переключения в состояние высокого импеданса) выхода `DDR`-триггера;
- `CLKFX_OUT` и `CLKFX180_OUT` — выходы сигналов с частотой 135 МГц, отличающихся по фазе на половину периода, которые формируются цифровым синтезатором частот модуля `DCM`;
- `CLKDV_OUT` — выход сигнала с частотой 15 МГц, формируемого делителем частоты модуля `DCM`;
- `CLKIN_IBUFG_OUT` — выход исходного тактового сигнала с частотой 90 МГц, снимаемого с входного буферного элемента;
- `CLK0_OUT` — выход сигнала с частотой 90 МГц с нулевым фазовым сдвигом, вырабатываемый модулем `DCM`;
- `CLK2X_OUT` и `CLK2X180_OUT` — выходы сигналов с частотой 180 МГц, отличающихся фазовым сдвигом, равным половине периода;
- `CLK90_OUT`, `CLK180_OUT` и `CLK270_OUT` — выходы сигнала с частотой 90 МГц с фазовым сдвигом на четверть, половину и три четверти периода соответственно;
- `DDR_CLKFX_OUT` — выход сигнала, вырабатываемого выходным `DDR`-триггером;
- `LOCKED_OUT` — выход сигнала, информирующего о соотношении фаз входного и выходного сигналов схем автоподстройки задержек `DLL` модуля `DCM`;
- `STATUS_OUT` — выходная шина состояния модуля `DCM`.

Установка значений всех необходимых атрибутов компонента `DCM_SP` осуществляется с помощью приведенной далее последовательности выражений, которая может быть включена в файл временных и топологических ограничений проекта разрабатываемого устройства в САПР серии Xilinx ISE.

```

INST DCM_SP_INST CLK_FEEDBACK = 1X;
INST DCM_SP_INST CLKDV_DIVIDE = 6.0;
INST DCM_SP_INST CLKFX_DIVIDE = 2;
INST DCM_SP_INST CLKFX_MULTIPLY = 3;
INST DCM_SP_INST CLKIN_DIVIDE_BY_2 = FALSE;
INST DCM_SP_INST CLKIN_PERIOD = 11.1111;
INST DCM_SP_INST CLKOUT_PHASE_SHIFT = NONE;
INST DCM_SP_INST DESKEW_ADJUST = SYSTEM_SYNCHRONOUS;
INST DCM_SP_INST DFS_FREQUENCY_MODE = LOW;
INST DCM_SP_INST DLL_FREQUENCY_MODE = LOW;
INST DCM_SP_INST DUTY_CYCLE_CORRECTION = TRUE;
INST DCM_SP_INST FACTORY_JF = C080;
INST DCM_SP_INST PHASE_SHIFT = 0;
INST DCM_SP_INST STARTUP_WAIT = FALSE;

```

Окончание следует